**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІ**

Кафедра **ПЗ**



### ЗВІТ

До лабораторної роботи № 1

**На тему:** *“ФОРМУВАННЯ ТЕХНІЧНОГО ЗАВДАННЯ ЯК РЕЗУЛЬТАТ АНАЛІЗУ ТА ВИЗНАЧЕННЯ ВИМОГ”*

**З дисципліни:** *“Вступ до інженерії програмного забезпечення”*

**Лектор:**

доцент каф. ПЗ

Левус Є. В.

**Виконав:**

ст. гр. ПЗ-15

Шпортко Т. О.

**Прийняв:**

асистент каф. ПЗ

Самбір А. А.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2023 р.

∑= \_\_\_\_ .

Львів – 2023

**Тема роботи:** Формування технічного завдання як результат аналізу та визначення вимог.

**Мета роботи:** Навчитися складати найпростіше технічне завдання до розробки програми.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**14)** Наведіть приклади нефункціональних вимог.

Нефункціональні вимоги - це вимоги, які не стосуються безпосередньо функцій системи, а описують якості або властивості системи, що повинні бути виконані. Ці вимоги визначають нелогічні аспекти системи, такі як безпека, надійність, ефективність, доступність, масштабованість, сумісність та інші аспекти.

Деякі приклади нефункціональних вимог:

* Надійність: система повинна бути стійкою до помилок і відмов.
* Безпека: система повинна бути захищеною від несанкціонованого доступу.
* Ефективність: система повинна бути швидкою та ефективною у використанні ресурсів.
* Доступність: система повинна бути доступною для користувачів у будь-який час.
* Масштабованість: система повинна бути здатною розширюватися зі зростанням навантаження.
* Сумісність: система повинна бути сумісною з іншими програмними засобами та обладнанням.
* Ергономіка: система повинна бути зручною у використанні та забезпечувати зручність користувача.

Нефункціональні вимоги дуже важливі для успішного розроблення та впровадження системи, оскільки вони визначають основні параметри якості та ефективності системи.

**33)** Які чинники визначають зручність користування інтерфейсом?

Існує безліч чинників, які визначають зручність користування інтерфейсом. Деякі з найбільш важливих чинників наступні:

* Простота та зрозумілість інтерфейсу: користувач повинен легко зрозуміти, як працювати з програмою або веб-сайтом.
* Наявність інструкцій та документації: інтерфейс повинен мати чіткі і зрозумілі інструкції для користувачів, якщо вони потребують допомоги.
* Повнота та точність інформації: інформація на інтерфейсі повинна бути повною, точною та актуальною.
* Наявність відповідних інструментів та функцій: інтерфейс повинен мати всі необхідні інструменти та функції, щоб користувач міг легко і ефективно виконувати завдання.
* Ефективність: інтерфейс повинен бути ефективним, щоб користувач міг швидко виконувати завдання та зменшувати кількість помилок.
* Відповідність очікуванням користувача: інтерфейс повинен відповідати очікуванням користувача щодо того, як він повинен взаємодіяти з програмою або веб-сайтом.
* Гнучкість та можливості налаштування: інтерфейс повинен бути гнучким і мати можливості налаштування, щоб користувач міг налаштувати його під свої потреби.
* Візуальна привабливість: інтерфейс повинен бути візуально привабливим та легким для очей користувача.
* Наявність підказок та сповіщень: інтерфейс повинен мати підказки та сповіщення, які допоможуть користувачеві швидко зорієнтуватися у виконанні завдань та запобігати помилкам.
* Наявність кнопок та інтерактивних елементів: інтерфейс повинен мати кнопки та інтерактивні елементи, які дозволять користувачеві виконувати дії з легкістю.
* Сумісність з різними пристроями та платформами: інтерфейс повинен бути сумісним з різними пристроями та платформами, щоб користувач міг використовувати його на будь-якому пристрої, на якому він працює.
* Стабільність та надійність: інтерфейс повинен бути стабільним та надійним, щоб користувач міг виконувати завдання без перебоїв та помилок.
* Наявність підтримки та обслуговування: інтерфейс повинен мати належну підтримку та обслуговування, щоб користувач міг отримати допомогу в разі потреби.

Зручність користування інтерфейсом залежить від багатьох факторів, і важливо забезпечити, щоб усі ці фактори були враховані при проектуванні інтерфейсу. Краще всього, якщо дизайн інтерфейсу розробляється з урахуванням потреб користувачів та з тестуванням на реальних користувачах.

**29)** У чому полягає складність отримання результатів аналізу й визначення вимог до ПЗ?

Складність отримання результатів аналізу та визначення вимог до програмного забезпечення (ПЗ) полягає в тому, що цей процес вимагає великої кількості знань і навичок з різних областей, а також взаємодії з багатьма зацікавленими сторонами.

Основні складності полягають в наступному:

1. Комплексність системи: ПЗ може включати в себе велику кількість складних елементів, що вимагає ретельного аналізу кожного елементу та їх взаємодії.
2. Різноманітність стейкхолдерів: Різні стейкхолдери (клієнти, користувачі, команди розробки, менеджмент) мають різні потреби і вимоги щодо ПЗ, і це може бути складно узгодити.
3. Змінність вимог: Вимоги до ПЗ можуть змінюватися з часом, тому процес аналізу та визначення вимог є постійним ітеративним процесом.
4. Технічна складність: Аналіз технічних аспектів ПЗ може бути складним, особливо якщо розробка базується на нових технологіях або включає в себе багато складних алгоритмів.
5. Недостатність інформації: У деяких випадках не вистачає достатньої кількості інформації про систему, що може ускладнити процес аналізу та визначення вимог.
6. Культурні та мовні бар'єри: Якщо стейкхолдери мають різну культуру або говорять різними мовами, то можуть виникати непорозуміння щодо вимог до ПЗ.

**ПОСТАНОВКА ЗАВДАННЯ**

Скласти технічне завдання (концептуальний рівень) до програми згідно індивідуального варіанту (№1-30) за поданим вище планом.

У п.4.3. представити зображення UI для демонстрації взаємодії користувача з програмою для кожної функції, зазначеної в п.4.1. Для забезпечення Usability використати евристики (UX) 1-8 (див. Додаток).

У п.4.4. задати дві нефункціональні вимоги.

У п.6 вказати документи, які будуть створені під час лабораторних робіт: блок-схеми алгоритмів, текст програми (за вимогами читабельності та зрозумілості), звіт про тестування.

У п.7 вказати два інформаційні джерела за темою програмування, як з Інтернет-ресурсів, так й друковані.

**Умова до програми:**

З клавіатури ввести послідовність записів, які містять дані про книгу : <Автор>, <Назва книги>, <Рік видання>, <Кількість сторінок>, <Вартість>. Роздрукувати введені дані у вигляді таблиці, а також подати інформацію згідно 3 варіанту. Передбачити зчитування вхідних даних з файлу та запис результатів у файл.

25. Відсортувати за назвою в алфавітному порядку дані про книги, вартість яких більша середньої в бібліотеці. Вилучити зі списку дані про книги з назвою, що починається на букви П, К, Л.

**ТЕХНІЧНЕ ЗАВДАННЯ**

***Пункт 1. - Загальні положення***

Найменування роботи: Книжкова база даних

Умовне позначення: BooksDB

Замовник: Левус Є. В.

Розробник: Шпортко Т. О.

Терміни:

* Початок робіт: 03.03.2023
* Закінчення робіт: 31.03.2023

**Пункт 2. - *Призначення системи***

* + Цілі створення програмного забезпечення:

Має за мету спростити облік книжковий облік, зменшити кількість помилок при отриманні, зберіганні та видачі екземплярів. Отже, має пришвидшити роботу цієї університетської установи.

* + Основні очікувані результати:

Має бути реалізоване подання даних про книжки у формі таблиці, яку зручно читати та редагувати, ключовими є можливості додавати нові книжки, видаляти непотрібні на вибір і за конкретною умовою (ті книжки, чия назва починається з літер «P», «K», «L»), сортувати в алфавітному порядку за назвою ([додатково]за умовою: ті книжки, що вищі середнього значення за ціною). Введення та виведення даних проводиться, працюючи з текстовим документом.

* + Область застосування даного продукту:

Дане ПЗ створюється для використання студентською бібліотекою НУ «Львівська політехніка». Можливе використання іншими ЗВО, ПЗ має бути сумісним для цього.

**Пункт 3. - *Об’єкти даних***

Інформація про книги подається у формі текстового документу.

(розширення .txt)

Він має чітко визначений формат:

Автор;Назва;Кількість сторінок;Рік створення;Ціна;

Містить у собі N (початково 17) таких рядків, заповнених книжковими даними.

* Назва книги та автор не перевищує довжини 50 символів, використовуючи тільки літери латинського алфавіту довільного регістру.
* Ціна є десятковим числом із точністю до сотих.
* Рік створення - ціле число, що містить максимум 4 розряди. Вказується як від’ємне число, якщо книгу написано до нашої ери.

Зразок запису в файл:

James Joyce;Ulysses;732;1922;8.76;

**Пункт 4. - *Вимоги до програмного забезпечення*.**

4.1. Функціональні вимоги:

* R1: Створення таблиці з даними про книжки розміром N\*5, де N-кількість книжок у базі даних (рядки), 5 – кількість їх характеристик (стовпці).
* R2: Можливість зчитування інформації з текстового файлу (розширення .txt) у книжковий список.
* R3: Можливість виводу книжкового списку у вигляді таблиці як у консоль, так і в текстовий або бінарний файл (розширення .txt чи .bin).
* R4: Можливість додавання книжок із введенням необхідних даних з клавіатури на задану позицію в таблицю.
* R5: Можливість видалення зі списку книжок, розташованих за заданою в таблиці позицією.
* R6: Сортування книжок в алфавітному порядку за назвою.
* R7: Можливість видалення зі списку тих книжок, чия назва починається з латинських літер P, K і L (довільного регістру).
* R8: Створення посортованого в алфавітному порядку списку з тих книжок, чия ціна вища за середню ціну наявних у базі даних.
* R9: Можливість виходу з програми.

4.2. Системні вимоги:

ОС: 64-bit Windows 7, 64-bit Windows 8 (8.1), 64-bit Windows 10, Unix-подібні

Процесор: Intel CPU Core i5-2500K 3.3GHz / AMD A10-5800K APU (3.8GHz)

Оперативна пам’ять: 4 GB ОП

Відеокарта: Nvidia GPU GeForce GTX 660 / AMD GPU Radeon HD 7870

Місце на диску: 275 MB доступного місця

4.3. Вимоги до користувацького інтерфейсу :

Перша версія продукту є консольним застосунком у формі інтерактивної таблиці. Передбачити підтримку графічного інтерфейсу в майбутньому.

Шрифт тексту: Times New Roman (наскреслення Bold чи Cursive), розмір шрифту: 14-22. Колір заднього фону - #000000, колір тексту - #ffffff. Колір верхньої панелі керування - #ffa500.

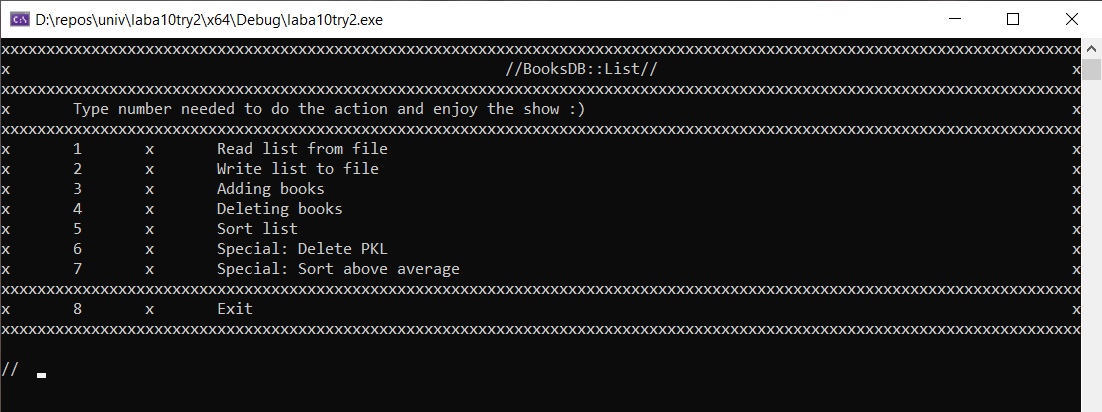
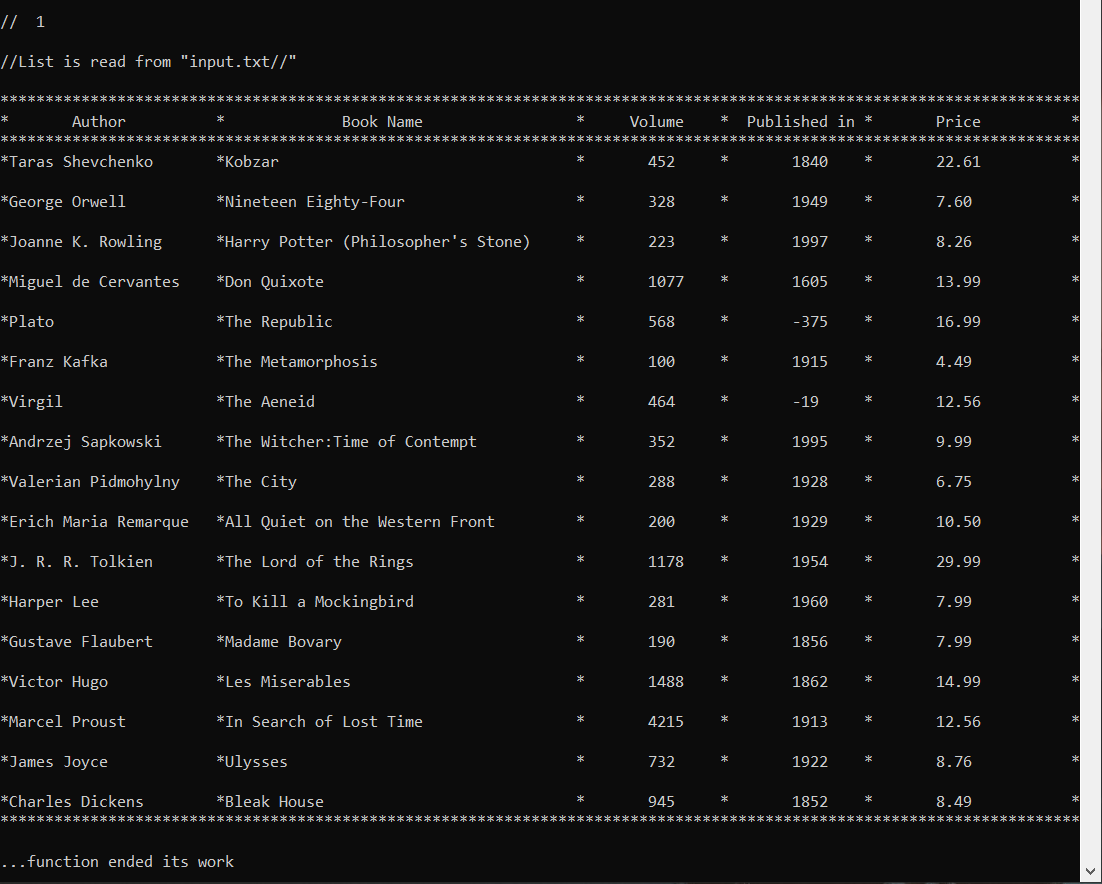


Рис. 1. Меню програми.



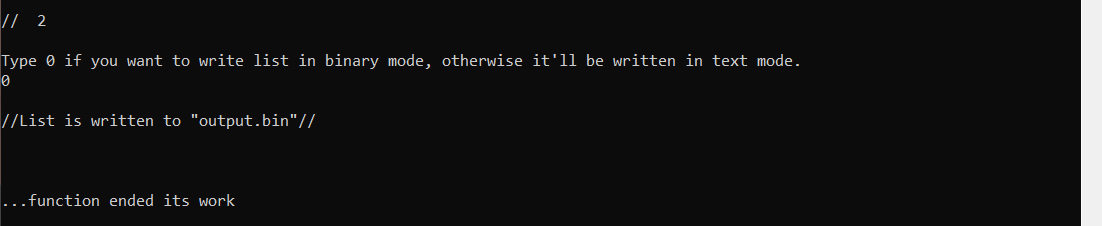
Рис. 2. Зчитування списку книг з файлу.

Рис. 3. Запис списку книг у файл.

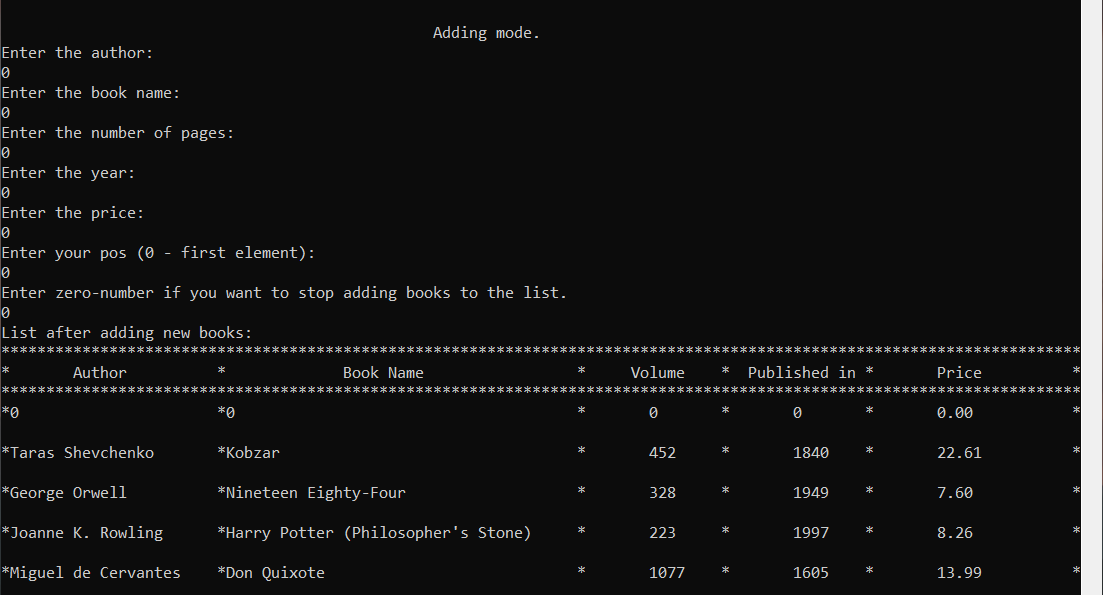
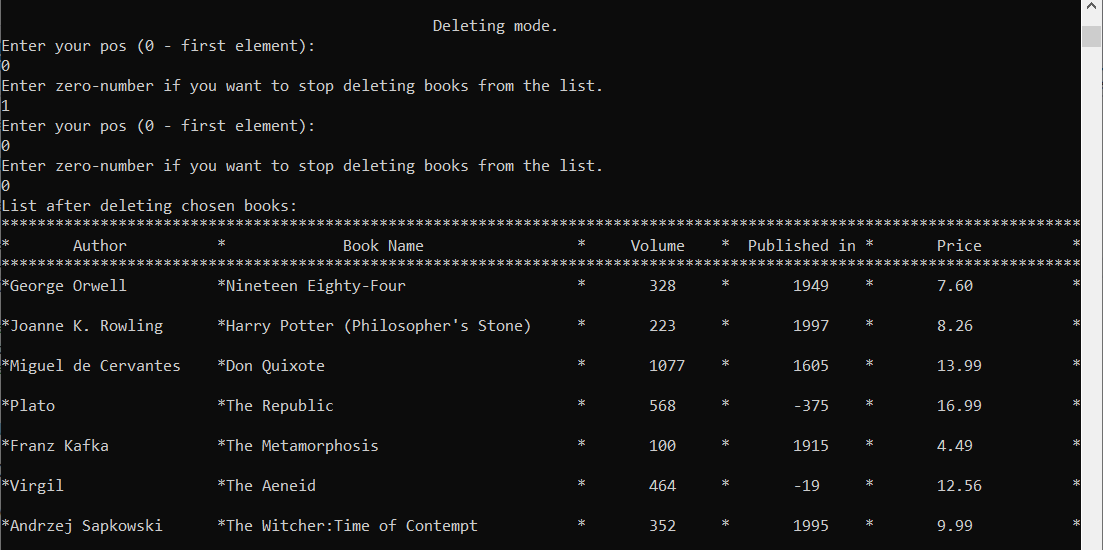
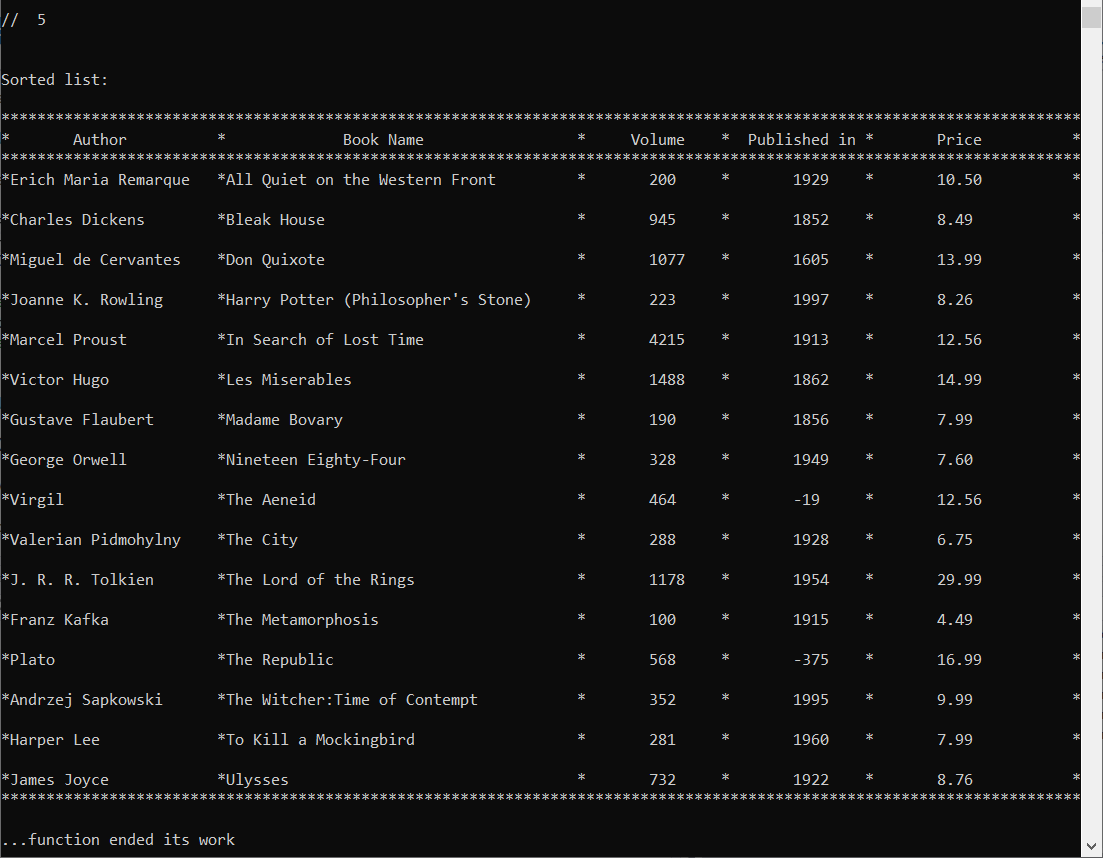
Рис. 4. Інтерфейс режиму додавання книг у список.

Рис. 5. Інтерфейс видалення книг зі списку.



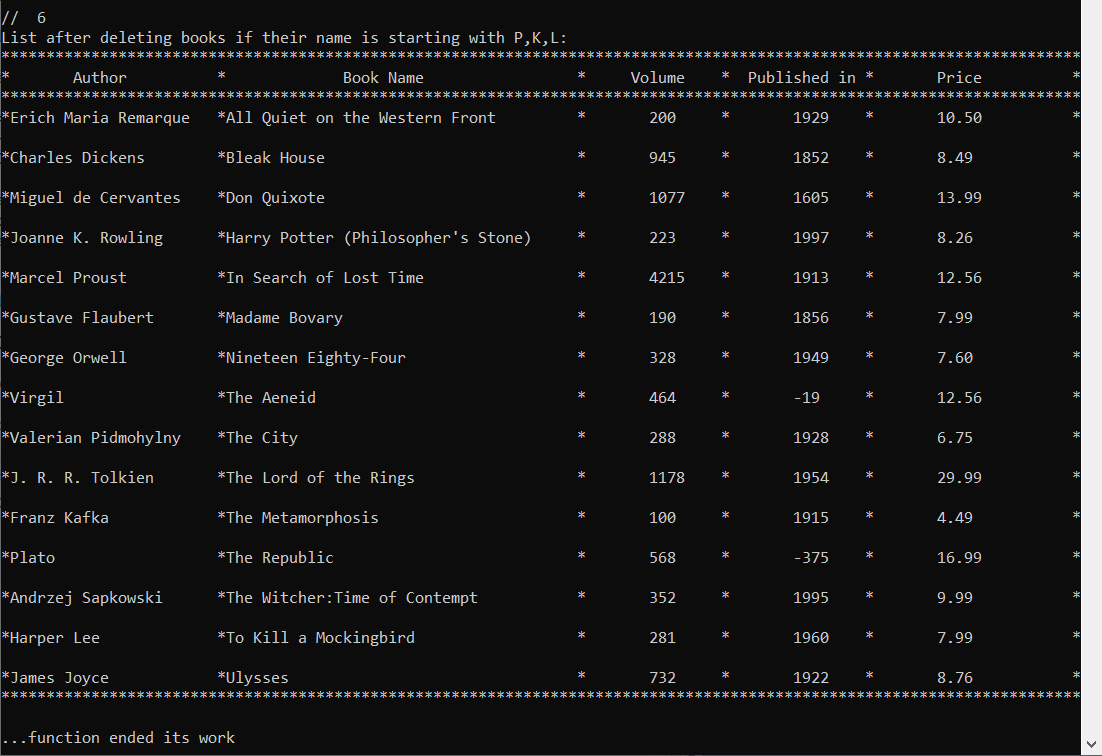
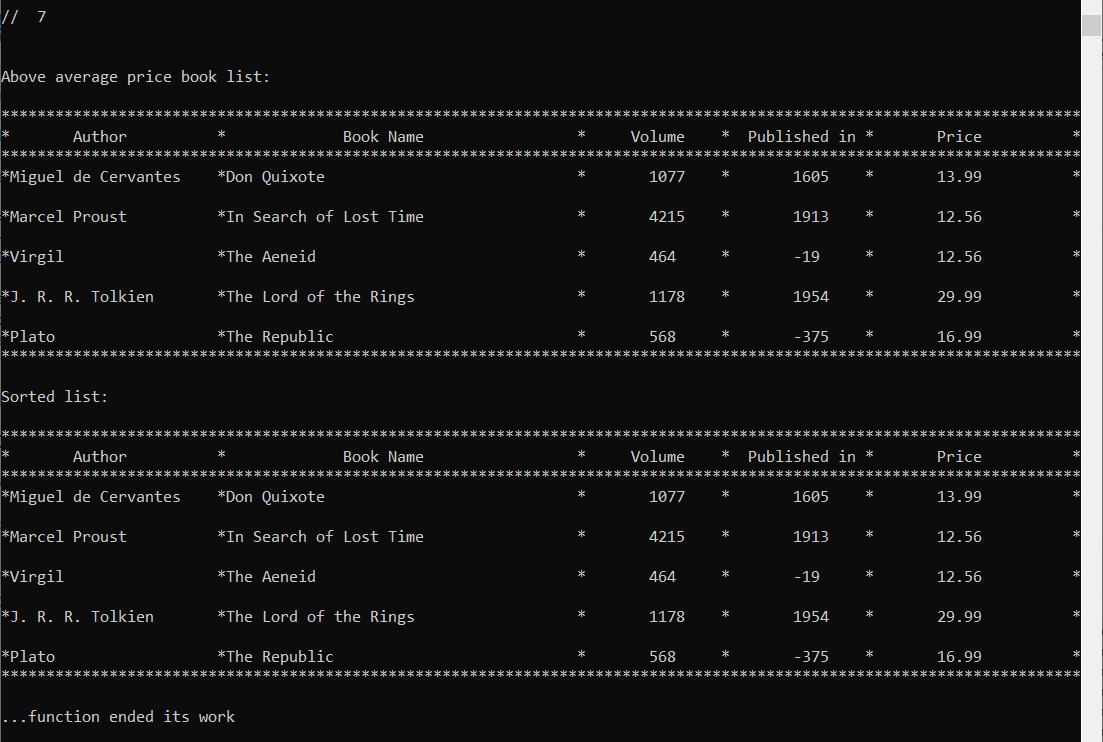
Рис. 6. Сортування списку книг в алфавітному порядку за назвою.

Рис. 7. Видалення зайвих книг зі списку (вимога R7).



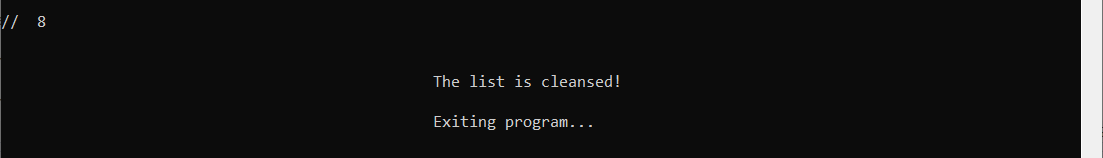
Рис. 8. Сортування книг вище середнього за ціною (вимога R8).

Рис. 9. Завершення роботи програми.

4.4. Інші нефункціональні вимоги

* Масштабованість: програма має підтримувати одночасне зберігання до 15000 книжок.
* Швидкість: незалежно від кількості інформації обробка кожного запиту користувача має займати не більше, ніж 5 секунд. Інтерфейс має реагувати миттєво.

**Пункт 5. - *Стадії розробки.***

* Підготовка
* Аналіз та специфікація вимог
* Проектування
* Кодування
* Тестування
* Додатково: створення документації
* Експлуатація
* Супровід
* Додатково: навчання персоналу

**Пункт 6. - *Вимоги до програмної документації.***

* Технічне завдання
* Блок-схеми алгоритмів й схематичне зображення структур даних
* Текст програми
* Звіти про тестування
* Прототипи графічного інтерфейсу користувача
* Інструкція для користувача

**Пункт 7. - *Посилання.***

* Левус Є. В. Життєвий цикл програмного забезпечення: навчальний посібник / Є. В. Левус, Т. А. Марусенкова, О. О. Нитребич. – Львів: Видавництво "Львівська політехніка", 2017. – 208 c.
* Шпак З. Я. Програмування мовою С. / Шпак З. Я. – Львів: Видавництво «Львівська політехніка», 2011. – 436 с.
* aCode / Онлайн-ресурс по програмуванню для початківців.

URL:

https://acode.com.ua/

**ВИСНОВКИ**

Виконуючи цю лабораторну роботу, я познайомився із життєвим циклом ПЗ, відповідно, його етапами, ретельно розглянув вимоги до ПЗ та до користувацького інтерфейсу, опрацював важливу складову взаємодії клієнта з розробником - склав просте технічне завдання. Складання ТЗ дозволяє визначити чіткий перелік вимог до програми та її функціональності, а також зробити оцінку ресурсів, необхідних для розробки програмного забезпечення, у результаті чого зменшується кількість помилок та непорозумінь при взаємодії клієнта з розробником.